LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-TR-491692

Comparison of Sequential Designs of Computer Experiments in High Dimensions

A. M. Kupresanin, G. Johannesson

August 1, 2011

Disclaimer

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Comparison of Sequential Designs of Computer Experiments in High Dimensions

A. Kupresanin and G. Johannesson

Abstract

We continue a long line of research in applying the design and analysis of computer experiments to the study of real world systems. The problem we consider is that of fitting a Gaussian process model for a computer model in applications where the simulation output is a function of a high dimensional input vector. Our computer experiments are designed sequentially as we learn about the model. We perform an empirical comparison of the effectiveness and efficiency of several statistical criteria that have been used in sequential experimental designs. The specific application that motivates this work comes from climatology.

1 Introduction

Experiments are often the most time-consuming and/or expensive component of modern scientific research. Therefore, in physics, chemistry, biology, etc. real experiments are often replaced by computational ones. A computational experiment consists of running a computer code that executes a mathematical model whose purpose is to simulate the real-world experiment. Examples of such models may involve the simulation of the behavior of a concrete or metal structure when under stress, the mechanics of a fluid (say, air flowing over an airplane wing), the energy storage in plant ecology, or a variety of other physical, chemical, biological and medical models. In certain cases, mathematical models allow a scientist to perform multiple experiments and analyze phenomena that could not even be reproduced in real world, for example questions related to weather conditions and climate. Most such computer experiments involve large numbers of input variables and are still fairly time-consuming. There are computer models whose individual runs require hours or days to complete. Since doing inference requires multiple input-output value pairs even when the basis is a computer model, completing a study in a reasonable amount of time requires an appropriate choice of the set of input conditions at which to run the model. The field of research addressing this problem is called design of computer experiments and has been growing rapidly. The fact that it is not possible to carry out computer simulations at very many input settings for code that runs slowly has led to the development of statistical methodologies for emulating the response of the code. One of the objectives of such work is to build an emulator (a.k.a. response surface) with good predictive accuracy.

Intuitively, in order to have a statistically relevant collection of input settings, one would like to have some settings near the boundary of the input space as well as in the interior. This idea has led to space-filling designs for computer experiments. Among the wide variety of such designs are Latin hypercube designs (LHD) [11], sphere packing designs, as well as others based on random or pseudo-random sequences of points. In most cases, these designs satisfy a property that says

that the design points are roughly uniformly scattered on the domain; such designs are said to be uniformly space-filling.

Another point of concern is that often, it is not clear at the outset how many points should be selected for a design. In contrast, the process of building a statistical model over a sequence of steps, by adding one or more points in each step, leads to improving accuracy of estimates. In some cases, it may be possible to add points to the design adaptively, using the already-built model to guide the selection of future points. This process may be called sequential, or adaptive design. In this paper, we compare several sequential design methodologies for response surface estimation.

Besides the procedures used to adaptively choose design points (that is, the adaptive sampling criteria), we also consider the more general question of the role the dimensionality (d) of the input space has in the design construction. If the curse of dimensionality applies, then the sample sizes required by high-dimensional problems will be massive and achieving good prediction accuracy will be an intractable problem. On the other hand, if the change of the code output due to changes in input variables is in some sense fixed, it may be possible that dimensionality has only a limited effect on the sample size needed to approach accuracy. How this "total responsiveness" of the simulation output grows with d and how it is spread across the input dimensions is the key to the understanding of the prediction accuracy achieved using a limited number of design points, as well as of the behavior of the various sampling criteria.

We keep to the path that has traditionally been taken in this area since 1989 [13], and use a set of code runs to build a Gaussian process (GP) in order to approximate the output of the computer experiments. In general, it is not possible to ignore the goals of the particular experiment of interest and they must be precisely formulated before factors affecting approximation quality can be characterized. Thus, we restrict our attention to the question of approximation and study the comparative behavior of several adaptive sampling criteria. In other words, we compare the various sampling criteria in terms of the improvement of prediction accuracy of the Gaussian process model on new inputs. The central question we ask is "Is there a criterion that clearly outperforms the others—and if not, can we draw some general conclusions about the criteria we study?"

One of the preliminaries to answering this question is to describe quantitatively the complexity of a GP model. In particular, the total responsiveness of an output variable to all the inputs is of major concern. The distribution of this responsiveness across the input variables is very important when d increases. The main conclusion of our case study is that, among the several natural sequential design criteria that we examine, there seems to be no clear winner in terms of global model fit.

The remaining part of the paper is organized as follows. Section 2 describes the computer model on which we base our study, the Community Atmosphere Model (CAM). In section 3 we review the GP model and explain the sequential sampling criteria that we use. We quantify the complexity of a GP in Section 4, where we also explore examples to illustrate the issues. Finally, in Section 5 we summarize our conclusions and comment on open issues.

2 Motivating example

In this section we discuss an example stemming from climate that motivated the present work. The Community Atmosphere Model (CAM) is a global atmosphere model developed by the weather and climate research communities. CAM also serves as the atmospheric component of the Community Climate System Model (CCSM). The details of the governing equations, physical parameterizations,

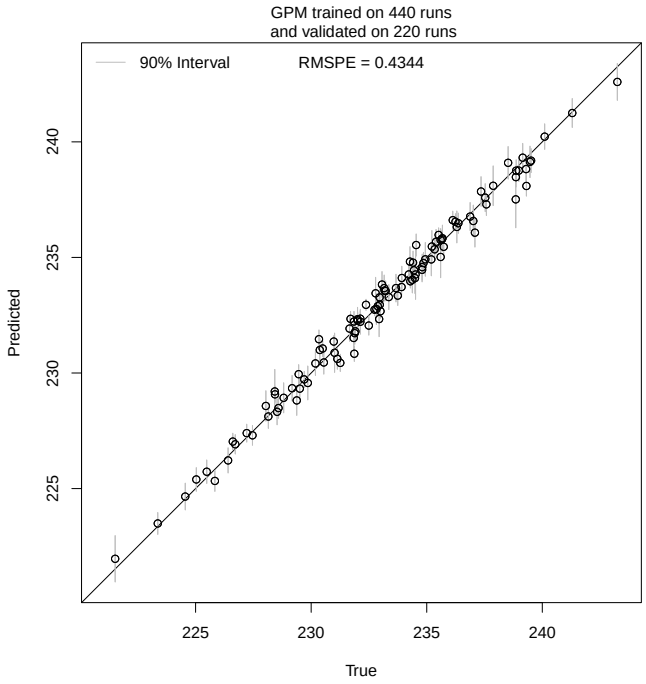

Figure 1: Validation plot for GP model of CAM. LHS of size 440 training, 220 validation points. The quantity plotted is the longwave radiation.

and numerical algorithms of CAM can be found in [2]. Among many uncertain input variables to CAM we consider 21 of them, selected by the climate research community. These 21 variables include cloud particle density, evaporation constant, cold ice autoconversion etc. CAM generates many output variables, some of them multivariate, as well as spatial-temporal fields. In this paper we simplify the output space and consider only a single scalar output, specifically the outgoing longwave radiation averaged over the entire globe and the last 10 years of the simulation. The computing time of one such run is 6 hours on approximately 400 cores. If one is attempting to reconstruct the response surface defined by the scalar output value, this computational load makes it critical that few runs that can be afforded are chosen carefully.

In Figure 1, we illustrate the performance of the Gaussian Process model for CAM by showing a validation plot of predicted (according to the GP model) versus true (according to CAM) values of longwave radiation average. To build the GP model, we used 440 CAM runs, with inputs selected according to a Latin Hypercube Sample (LHS) [11]. Details of the GP model are given in Section 3. The plot consists of 220 points at which the statistical model output was compared to that of CAM. These 220 validation points were selected the same way as the 440 training points, that is according to an LHS. The validation plot shows that, at least for this setting, the performance of the GP model quite closely matches the outputs obtained from CAM runs.

3 Surrogate model and adaptive sampling

In our setting, the results of experiments, that is, outputs of the computer code runs, are deterministic because CAM code is deterministic, and so any lack of fit is not due to noise but entirely to modeling error. Complex computer code outputs are frequently modeled by Gaussian processes. Possibly the major reason for this is that GP models may be constructed that fit a broad class of potential response surfaces—in any case a much broader class than the more typically used polynomial models.

3.1 Gaussian Process Model

The heavy demands of complex computer codes have forced the development of computationally efficient strategies that involve statistical prediction at least since the work of Sacks [13] in 1989. Such strategies are often referred to also as emulation or approximation of the underlying code. We follow such work in placing a homogeneous GP prior on the family of possible output functions. Thus, our predictor will be given by the posterior mean conditional on the output of the computer experiment. In the current work, we focus on the case of scalar output even though modeling a multivariate is also of great interest.

We use $y(\mathbf{x})$ to denote the output of the code, where the vector input variable, $\mathbf{x} = (x_1, \ldots, x_d)$, is restricted to the d-dimensional unit cube. This assumption is easily relaxed to any "rectangular" input space. The GP model then places a prior on the class of possible functions $y(\mathbf{x})$. We denote by $Y(\mathbf{x})$ the random function whose distribution is determined by the prior. Suppose that $Y(\mathbf{x}) = \mu + Z(\mathbf{x})$, where μ is a mean parameter and $Z(\mathbf{x})$ is a Gaussian stochastic process with mean 0, constant variance σ^2, and correlation function given by

$$R(\mathbf{x}, \mathbf{x}') = exp(-h(\mathbf{x}, \mathbf{x}')), \tag{1}$$

where

$$h(\mathbf{x}, \mathbf{x}') = \sum_{j=1}^{d} \theta_j |x_j - x_j'|^{p_j}, \tag{2}$$

with $\theta_j \geq 0$ and $1 \leq p_j \leq 2$. Here, $\mu, \sigma^2, \theta_j, p_j$ are the parameters of the prior. We estimate the values of these parameters from the outputs of the computer model runs and then "plug them into" the formula for the distribution of $Y(\mathbf{x})$ (an empirical Bayes approach [12]).

In situations where the output is quite smooth, it is often the case that $p_j = 2$ holds for all j, resulting in a Gaussian correlation function. For the remainder of this article, we make this assumption and proceed to focus on this special case. Our approach to the understanding what affects the prediction accuracy (and hence the performance of the various sampling criteria) is to study and try to characterize the distribution of the correlation values $R(\mathbf{x}, \mathbf{x}') = exp(-\sum_{j=1}^{d} \theta_j |x_j - x_j'|^2)$ between different design points as a function of the value of the parameters $\theta_1, \ldots, \theta_d$. One may interpret θ_j as a measure of the "responsiveness" of $Y(\mathbf{x})$ to x_j.

Consider running the code at the n input vectors $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)} \in [0, 1]^d$. Denote the data (the vector of the respective outcome values) by $\mathbf{y} = (y(\mathbf{x}^{(1)}), \ldots, y(\mathbf{x}^{(n)}))'$. Then the predictor $\hat{Y}(x)$ for $Y(\mathbf{x})$ is the posterior mean of $Y(\mathbf{x})$ given the data and $\theta = (\theta_1, \ldots, \theta_d)$,

$$\hat{Y}(\mathbf{x}) = E\left(Y(\mathbf{x}) | \mathbf{y}, \theta\right) = \hat{\mu} + \mathbf{r}'(\mathbf{x})\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}), \tag{3}$$

where $\mathbf{r}(\mathbf{x}) = (R(\mathbf{x}, \mathbf{x}^{(1)}), \ldots, R(\mathbf{x}, \mathbf{x}^{(n)}))'$ is an $n \times 1$ vector of correlations as in (1), $\mathbf{R}$ is an $n \times n$ matrix whose (i, j)-entry is given by $R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$. Here, $\hat{\mu}$ is the maximum likelihood estimate of μ and $\mathbf{1}$ is an $n \times 1$ vector whose all elements are equal to 1. The mean squared error (MSE) of $\hat{Y}(\mathbf{x})$, taking into account the uncertainty from estimating μ by maximum likelihood, is given by

$$MSE\left(\hat{Y}(\mathbf{x})\right) = E\left(\hat{Y}(\mathbf{x}) - Y(x)\right)^2 = \sigma^2\left(1 - \mathbf{r}'(\mathbf{x})\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}) + \frac{\left(1 - \mathbf{1}'\mathbf{R}^{-1}\mathbf{r}(\mathbf{x})\right)^2}{\mathbf{1}'\mathbf{R}^{-1}\mathbf{1}}\right). \quad (4)$$

In practice, σ^2 and θ also must be estimated. For this, we rely on the maximum likelihood estimate (see [14], equation 3.3.14). The MSE in (4) is the Bayes posterior variance conditional on plugging the covariance parameters.

3.2 Emulator guided adaptive sampling criteria

In this section we discuss sequential design criteria that may be used with the GP model and define several such criteria that we will study. We note some important differences in the design strategies for computer experiments, as opposed to those used when dealing with traditional (physical) experiments:

1. The output of the computer code is deterministic, and so the same input will yield the same response, regardless of the number of runs made on it. Therefore, a design should not take more than one observation at any input point.

2. A design should be flexible enough so that the responses collected at the sampled points provide sufficient information to reconstruct the functional form of the response surface on the entire input space. Since the functional relationship between the input and the output is usually not known, it is not known where to focus the samples (for example, the regions where the response varies significantly) and thus the natural way to interpret this requirement is as implying that the design be space-filling.

Experimental designs relevant to computer experiments are often broadly categorized into two classes: space-filling and criterion-based designs. According to item **2.** above, in order to achieve good prediction accuracy, and to minimize the overall prediction error of the GP model, it is natural to consider a space-filling design strategy. Several examples of space-filling designs have been studied. They include methods based on selecting samples randomly, e.g. Latin hypercube designs (LHD), distance-based designs and uniform designs. Thorough discussions of the various strategies may be found, for example, in Satner et al. [14], Koehler and Owen [7] and Bates et al. [1]. Latin hypercube sampling dates back to McKay at al. [11] and was introduced as an alternative to simple random sampling and stratified sampling. The intuition behind its definition is to ensure that the input points are uniformly distributed over the range of each input dimension. In our setting, space-filling designs are great for initial exploratory analysis, but their general applicability is limited by their very construction rationale, that is, the asumption that the interesting features of the response surface are equally likely to be found anywhere in the input space. Thus, if we follow a space-filling design blindly, we will have no freedom to adapt our selection of input points to what we learn about the response surface as the model is built point by point. If we use the knowledge recovered from the partially constructed model, we may be able to adaptively select the samples to focus on regions of the input space where more interesting features appear and

thus improve greatly over a pure space-filling design's prediction accuracy and efficiency. This reasoning leads to the second broad class of designs: those constructed by adding one or several points at a time to the initial design, with the choice of new points based on statistical criteria and an evaluation of the existing (partial) design, rather than the purely geometric criteria used in space-filling designs. Due to the way they are built up piecemeal, these designs are usually referred to as sequential, or adaptive. Sequential designs based on optimizing statistical criteria such as mean squared prediction error or the notion of entropy have also been used to construct designs for computer experiments [14]. However, they are usually difficult to implement because of their dependance on the unknown correlation parameters present in the GP model.

Traditionally, designs for modeling computer experiments have been almost exclusively restricted to LHDs which appears to be mainly due to the wide availability of software that generates them efficiently, even when the input is high-dimensional. We believe that sequential designs can be a significantly more effective and efficient prediction tool than fixed point designs if they are adaptive, i.e., the GP model is updated sequentially so that new design points are added based on the information recovered from the previous iteration of the model about the features of the response surface under approximation.

3.2.1 Maximum mean squared prediction error

The mean square error (4) of the best linear unbiased predictor (3) is a measure of the prediction uncertainty of the GP model and can be used as a design criterion. To implement this criterion, the new input point $\mathbf{x}_0$ should be selected so as to maximize the MSE based on the constant mean GP model fitted using the existing input points:

$$\max_{\mathbf{x}_0} MSE(\mathbf{x}_0) = \max_{\mathbf{x}_0} \sigma^2 \left(1 - \mathbf{r}'(\mathbf{x}_0)\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}_0) + \frac{\left(1 - \mathbf{1}'\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}_0)\right)^2}{\mathbf{1}'\mathbf{R}^{-1}\mathbf{1}} \right). \tag{5}$$

Since the correlation $r(\mathbf{x}_0)$ decreases with increasing distance between two input points, the maximum MSE design tends to spread out points and, in particular, the initial points will tend to lie on the boundaries of the input space. Unless important features of the true response surface are on or near the boundary, the surface thus fitted may be a poor approximation until the number of design points is large enough to also adequately cover the interior of the design region.

3.2.2 Maximum entropy

As described above in the motivation for using sequential designs, using the information provided by the partial model is of great importance. However, besides the information itself, it may be of interest to consider the amount of information provided by a candidate point. The notion of entropy is standard in information theory, and it turns out that it may also be used as a sequential design criterion. In 1987, Shewry and Wynn [17] proposed sampling by maximum entropy when the design space is discrete and showed that the expected change in the amount of information provided by an experiment is maximized by the design that maximizes the entropy of the observed responses. Their definition of entropy follows that of measure of information introduced in Lindley [9], as well as Shannon's Entropy [16]. Currin et al. [3] applied this idea to the design of computer experiments. It can be shown [7] that the maximum entropy design maximizes the determinant of the observation covariance matrix in (1). The maximum entropy design criterion can also be adapted for use as a

sequential algorithm. The correlation matrix $\mathbf{R}$ which now includes the candidate point $\mathbf{x}_0$, can be partitioned into

$$\mathbf{R} = \begin{pmatrix} \mathbf{R}_n & r_n(\mathbf{x}_0) \\ r_n'(\mathbf{x}_0) & 1 \end{pmatrix} \tag{6}$$

where $\mathbf{R}_n$ is the correlation matrix based on the existing n design points only. The cross correlation between the observation at a new candidate point $\mathbf{x}_0$ and observations at the existing design points is denoted by the vector $r_n(\mathbf{x}_0)$. As a result, $\det(\mathbf{R})$ can be written as a product of $\det(\mathbf{R}_n)\left(1 - r'(\mathbf{x}_0)\mathbf{R}_n^{-1}r(\mathbf{x}_0)\right)$. This is a product of scalar terms and the sequential maximum entropy criterion reduces to selecting a new point that satisfies

$$\max_{\mathbf{x}_0} \left(1 - r'(\mathbf{x}_0)\mathbf{R}_n^{-1}r(\mathbf{x}_0)\right). \tag{7}$$

3.2.3 Maxmin cross validation

Cross validation provides another alternative to MSE as a measure of prediction error for the stochastic model specified in (4). This measure of prediction error will be directly used as part of a sequential design as the criterion in selecting additional input points. To motivate this approach, observe that the MSE of the model depends directly on the distances between sampled input points $\mathbf{x}$ and on the correlation function $\mathbf{R}(\cdot)$, but also indirectly on the response values observed at these input points—that is, on the predicted values given by the fitted surface. By considering criteria based on cross validation, we use the observed and predicted responses.

Let $\mathbf{x}_0$ denote a candidate point and $\hat{Y}^{(-j)}(\mathbf{x}_0)$ denote the best linear unbiased predictor (BLUP) of $y(\mathbf{x}_0)$ based on all the data except $\{\mathbf{x}_j, y(\mathbf{x}_j)\}$ where $\{\mathbf{x}_j : j = 1, \ldots, n\}$ are the sampled points, while $\hat{Y}(\mathbf{x}_0)$ denotes the BLUP of $y(\mathbf{x}_0)$ using all the data. To reduce the computational burden, the correlation parameters for the BLUP are estimated based on all n observations. The maxmin cross validation prediction error criterion requires us to compute the minimum cross validation prediction error for every candidate point and choose as the next point the one with the largest error. That is, we maximize

$$CV(\mathbf{x}_0) = \min_j \left(\hat{Y}^{(-j)}(\mathbf{x}_0) - \hat{Y}_n(\mathbf{x}_0)\right)^2. \tag{8}$$

The cross validation approach has been studied by Jin et al. [4] and Kleijnen and Beers [6].

3.2.4 Maximum expected improvement

The expected improvement (EI) criterion was proposed by Schonlau [15] and originally developed in the context of global optimization by Jones et al. [5]. Lam [8] considered a modification of this criterion with the goal of obtaining a good global fit of the GP model instead of locating the global optimum or optima. Intuitively, the objective here is to search for "informative" regions in the domain that will help improve the global fit of the model, where "informative" means regions with significant variation in the response variable (compare this to the maximum entropy criterion). Following the notation of [8], suppose we have the computer experiment outputs $y(\mathbf{x}_j)$ at sampled points $\mathbf{x}_j$, $j = 1, \ldots, n$. For each potential input point $\mathbf{x}_0$, define its improvement as

$$I(\mathbf{x}_0) = \left(Y(\mathbf{x}_0) - y(\mathbf{x}_{j^*})\right)^2, \tag{9}$$

where $y(\mathbf{x}_{j^*})$ is the observed output at the sampled point $\mathbf{x}_{j^*}$ closest (in Euclidian distance) to the candidate point $\mathbf{x}_0$. The maximum expected improvement criterion advises to select as the next point the one that maximizes the expected improvement

$$E\left(I(\mathbf{x}_0)\right) = \left(\hat{Y}(\mathbf{x}_0) - y(\mathbf{x}_{j^*})\right)^2 + var\left(\hat{Y}(\mathbf{x}_0)\right). \tag{10}$$

For the details of the derivation of $E(I)$, we refer to [8].

The estimate of expected improvement in (10) uses two search components, one local and one global. The first (local) component of the expected improvement will tend to be large at points where the increase in response over the nearest sampled point is large. The second (global) component is large at points with the largest prediction error as defined in (4), i.e., points about which there is large uncertainty (these tend to be far from existing sampled points).

4 Examples

4.1 Challenges of generating synthetic data

The central question in our case study is the comparison among the efficiency of several adaptive sampling criteria. By efficiency, we mean the rate at which the method based on a criterion achieves some acceptable accuracy as a function of the number of samples. We would also like to be able to understand how this depends on the dimensionality of the input space. Since our real data set is limited both by its extreme computational demands and its fixed dimensionality, we need to be able to synthetically generate multiple data sets with arbitrary dimensionality of input space, while still keeping the computational requirements low.

The CAM application establishes that simulation from a GP model may have properties at least close to mimicking reality. But this is just one example, and we would like to know whether different adaptive sampling criteria provide different levels of prediction accuracy, and to study this question across a wider class of functions of higher dimensionality and greater complexity. The effects of dimensionality, d, correlation parameters, θ, and sample size n have already been discussed by Loeppky et al. [10]. For completeness, we first review their findings.

One of the problems we face is the generation of synthetic data where the correlation parameters scale across dimensions. Intuitively, the design points will tend to lie closer to each other as n grows larger, leading (by continuity) to improved accuracy in the prediction of $Y(\mathbf{x})$. However, if θ has many components with large values, then the correlation between $Y(\mathbf{x})$ and Y for the neighbors will be low, even for nearby points, leading to poorer prediction accuracy. Loeppky et al. [10] develop this intuition into a quantitative rule relating d, θ, and n to distances between input points and their correlation structure. For fixed n, they derive the mean and variance of the distribution of the distances in (2) between design points. This leads to a characterization of the distribution of correlations in $\mathbf{R}$ and the distribution of correlations in $\mathbf{r}(\mathbf{x})$ for $\mathbf{x}$ drawn randomly from $[0,1]^d$. They conclude that the mean and variance of the distance distribution explain much of the effect of d on θ.

Motivated by these thoughts, we address the problem of scaling across dimensions by generating synthetic data from a Gaussian process so that at the expected distance between two randomly selected points, the correlation is fixed. More precisely, we start with the observed correlation parameters of the Gaussian process model fitted to CAM data (Figure 2) and generate samples

from a Gaussian process in 2, 4, 8, 16 and 32 dimensions, respectively, so that the shape of the correlation parameter sequence of each of these models mimics that estimated from CAM data.

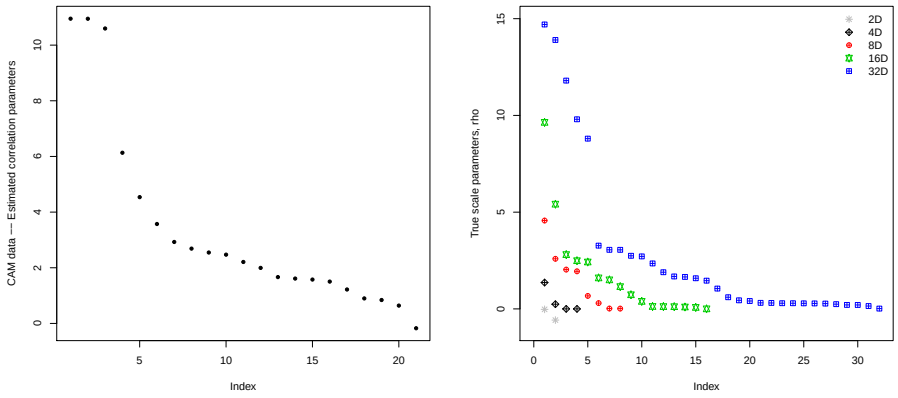

Figure 2: CAM data estimated correlation parameters (*left*) GP model correlation parameters for dimension 2,4,8,16 and 32 (*right*)

Dimension	Training data	Candidate data	Validation data
2	25	1000	1000
4	50	2000	2000
8	100	4000	4000
16	200	8000	8000
32	400	16000	16000

Figure 3: Synthetic data properties. Training data refers to the number of points used to train the Gaussian process model. Then, adaptive sampling criteria are used to select a certain number of additional points from the candidate pool. By experimentation, we settled on the rule of adding 10 points at a time, until either a conclusion about the performance could be reached, or the computational resources were exceeded. Finally, the resulting models were evaluated using the validation pool.

We base our findings on three families of data sets, and illustrate the results on one example of each. These are:

- stationary Gaussian process with low noise
- stationary Gaussian process with high noise
- nonstationary Gaussian process

The figures in the next section show the performance of the adaptive sampling criteria on all the cases described above. For comparison, in each of the cases, we also ran several experiments nonadaptively, by adding new points to the model randomly from the pool of candidate points. Of course, we followed the standard regimen of adding 10 points at a time. Several of these random models were generated in each case, and their outcomes are shown next to the results of adaptive criteria.

4.2 Synthetic data from stationary GP model

We use Gaussian processes to model computer code behavior, but in general, we do not expect the fit to be perfect. In the best imaginable case, the computer code output would be a draw from a Gaussian process. We simulate this best case scenario, generating synthetic data, and use this to evaluate the predictive performance of the various designs. Figure 5 indicates that overall, sequential designs tend to do better than random selection of points. This is the case even as the dimension increases. However, confirming our expectations, in the scenario when the noise in the data is high and the initial estimate of the response surface does not have good predictive performance (Figure 6), sequential addition of future runs according to a design criterion does not improve the prediction over random selection of points (Figure 7).

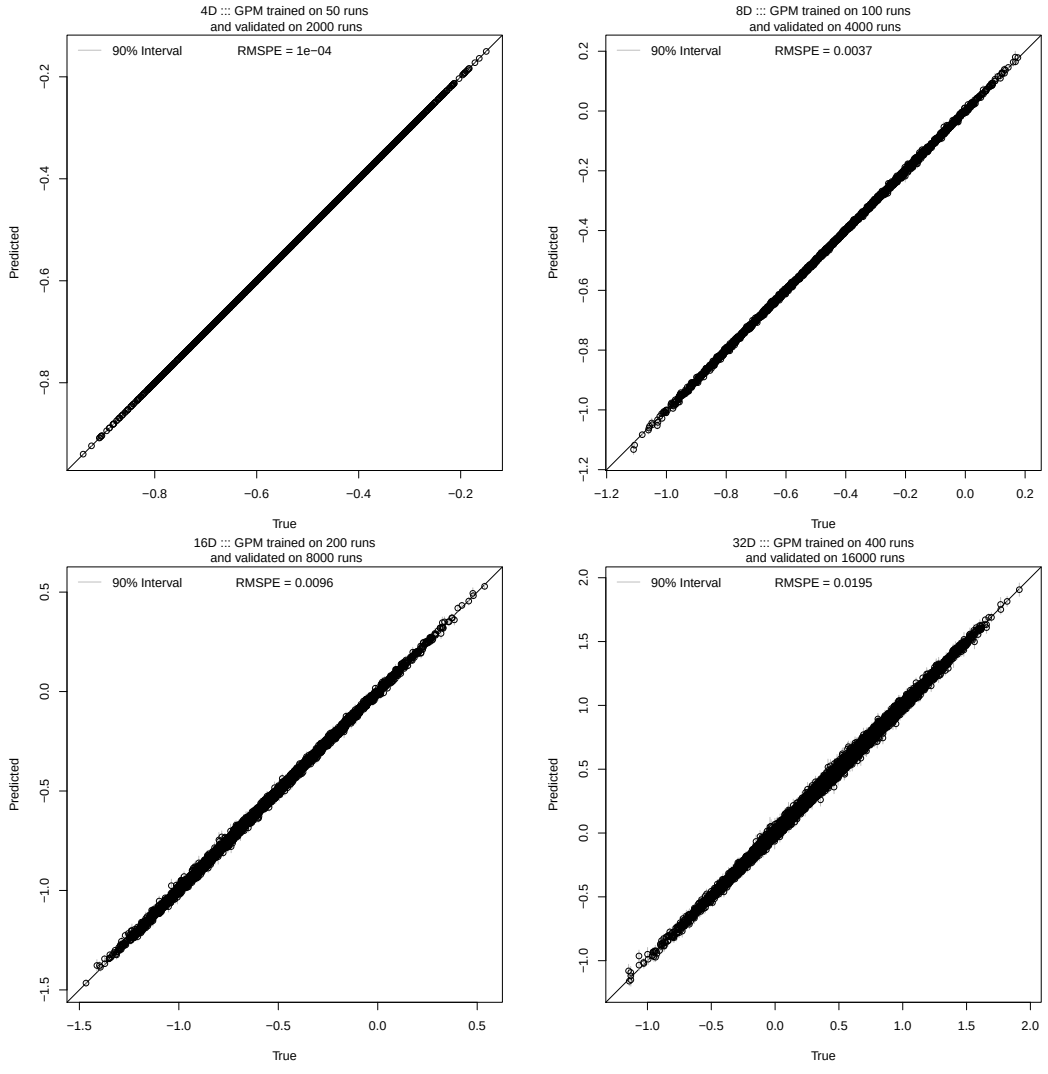

Figure 4: Validations plots of predicted (according to the GP model) versus true values for synthetic data generated from the stationary GP model in 4, 8, 16 and 32 dimensions. The GP model fits the data well due to the very low noise level in the data.

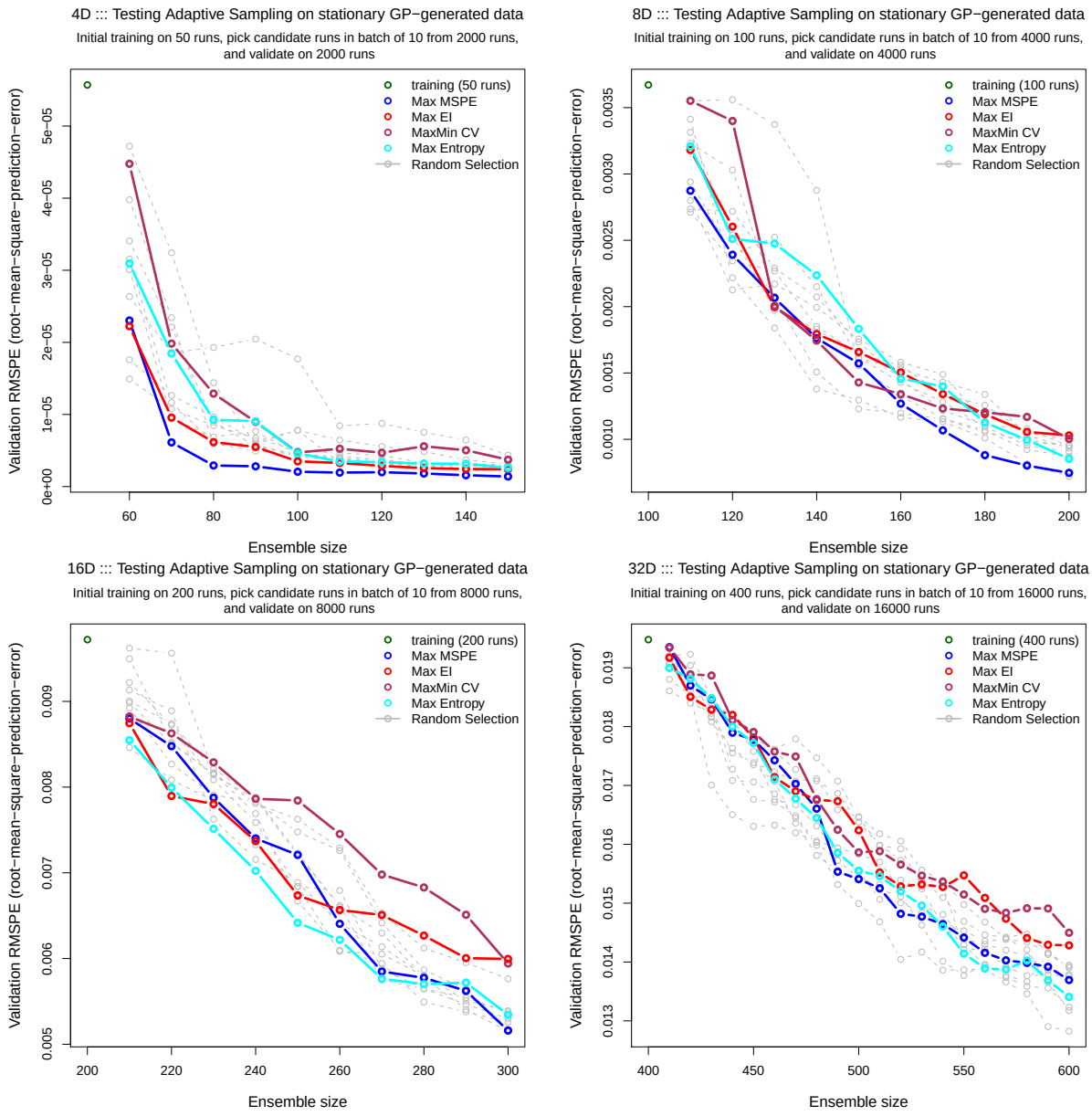

Figure 5: Performance of the adaptive sampling criteria across dimensions 4, 8, 16 and 32 in case when the data are generated from a stationary GP model with a low noise level. The GP is trained on the initial set of runs and a new set of 10 runs is added to the training set according to the four described criteria as well as randomly from a candidate set of runs. The root mean squared prediction error is calculated by comparing to the validation data set.

4.3 Synthetic data from non-stationary GP model

Our next set of results concerns a synthetic data set drawn from a nonstationary Gaussian process which is then modeled by a stationary Gaussian process. The reason for this is that in practice the actual distribution of the data is not known. Thus, a frequently used approach is to just use a

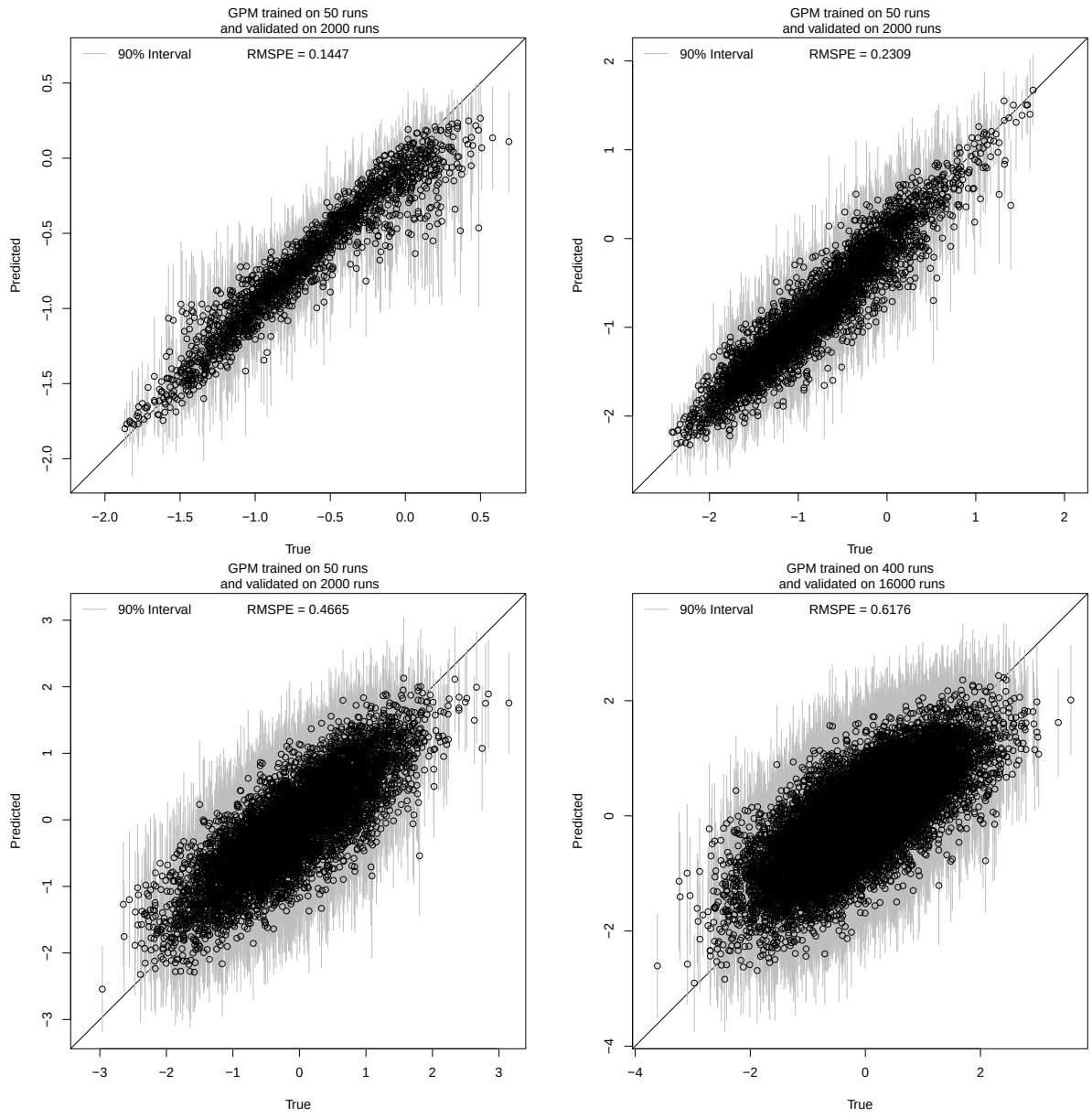

Figure 6: Validations plots of predicted (according to the GP model) versus true values for synthetic data generated from the stationary GP model in 4, 8, 16 and 32 dimensions. There is a high noise level in the data and the GP does not fit the data well. This becomes more apparent as the dimensionality increases.

GP model with a stationary covariance function (in which the covariance between any two points is a function of Euclidean distance). We want to illustrate that this naive approach of specifying a single stationary GP model across the entire input space of a clearly nonstationary response need not suffer in terms of prediction as long as the design criterion is able to target regions with high variation in the response. Although no one criterion stands out across all the dimensions and data

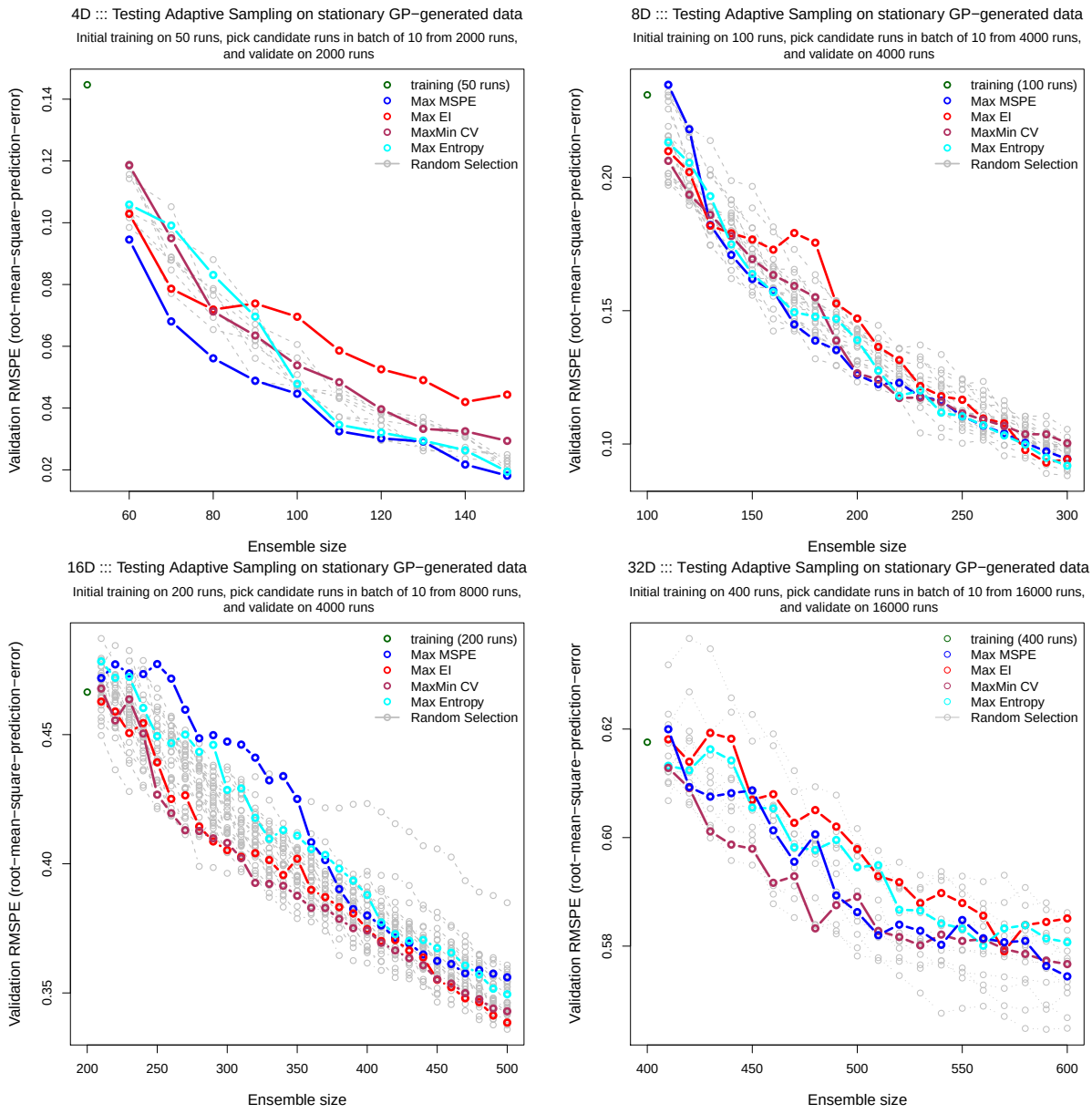

Figure 7: Performance of the adaptive sampling criteria across dimensions 4, 8, 16 and 32 in case when the data are generated from a stationary GP model with a high noise level. A stationary GP model is trained on the initial set of runs and a new set of 10 runs is added to the training set according to the four described criteria as well as randomly from a candidate set of runs. The root mean squared prediction error is calculated by comparing to the validation data set.

sets we examined, Figure 9 indicates that the EI criterion seems to perform better in predicting nonstationary looking response surfaces.

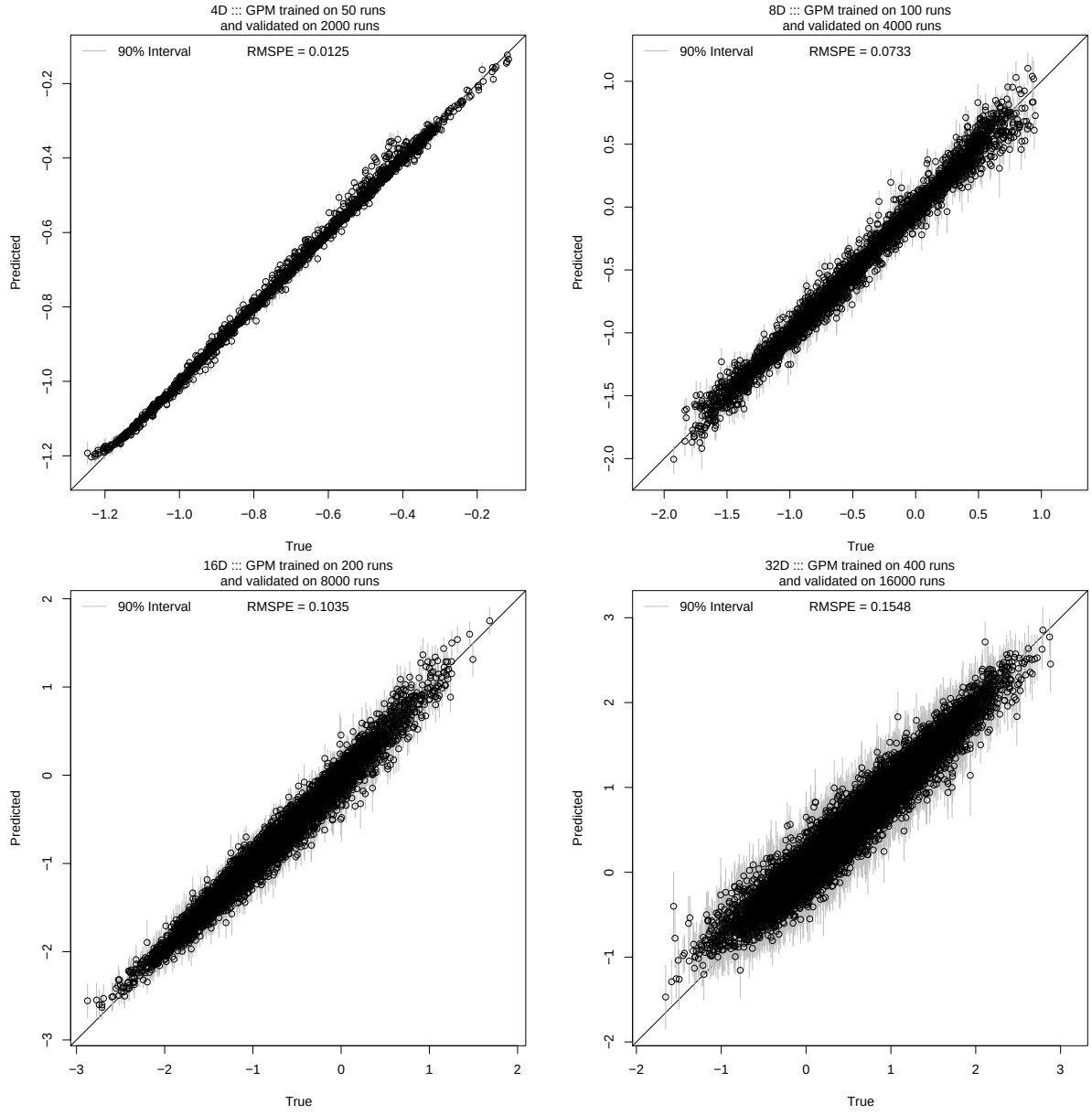

Figure 8: Validations plots of predicted (according to the stationary GP model) versus true values for synthetic data generated from the nonstationary GP model in 4, 8, 16 and 32 dimensions. The synthetic data are generated from a nonstationary GP model and then a stationary GP model is fit to the data. The plots show a decrease in the prediction accuracy with increase in the dimensionality.

4.4 CAM data

For comparison to synthetic data, we performed a similar regimen for each of the adaptive sampling criteria on actual CAM data. Of course, due to the complexity of the problem, all data pools were smaller. Namely, 120 runs were used to train the initial Gaussian process model. Then 10

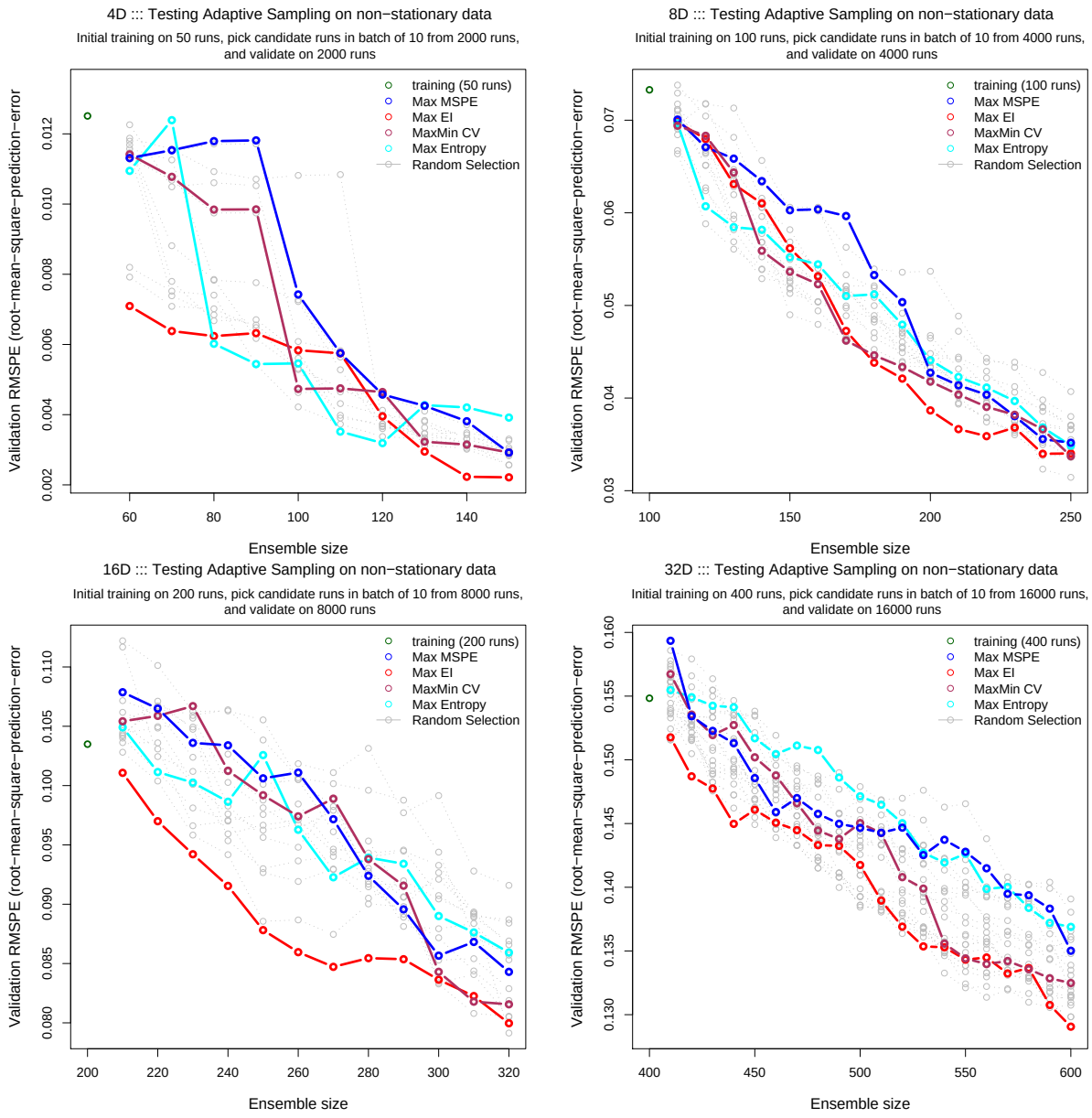

Figure 9: Performance of the adaptive sampling criteria across dimensions 4, 8, 16 and 32 in the case when the data are generated from a nonstationary GP model. The GP model is trained on the initial set of runs and a new set of 10 runs is added to the training set according to the four described criteria as well as randomly from a candidate set of runs. The root mean squared prediction error is calculated by comparing to the validation data set.

additional points were added at a time, from a pool of 440. Finally, validation was performed using 110 further runs of CAM. The results are shown in Figure 10. It appears that the Max MSPE criterion outperforms all others. We attribute this to the fact that only a few of the 21 input dimensions we consider are actually active in this problem.

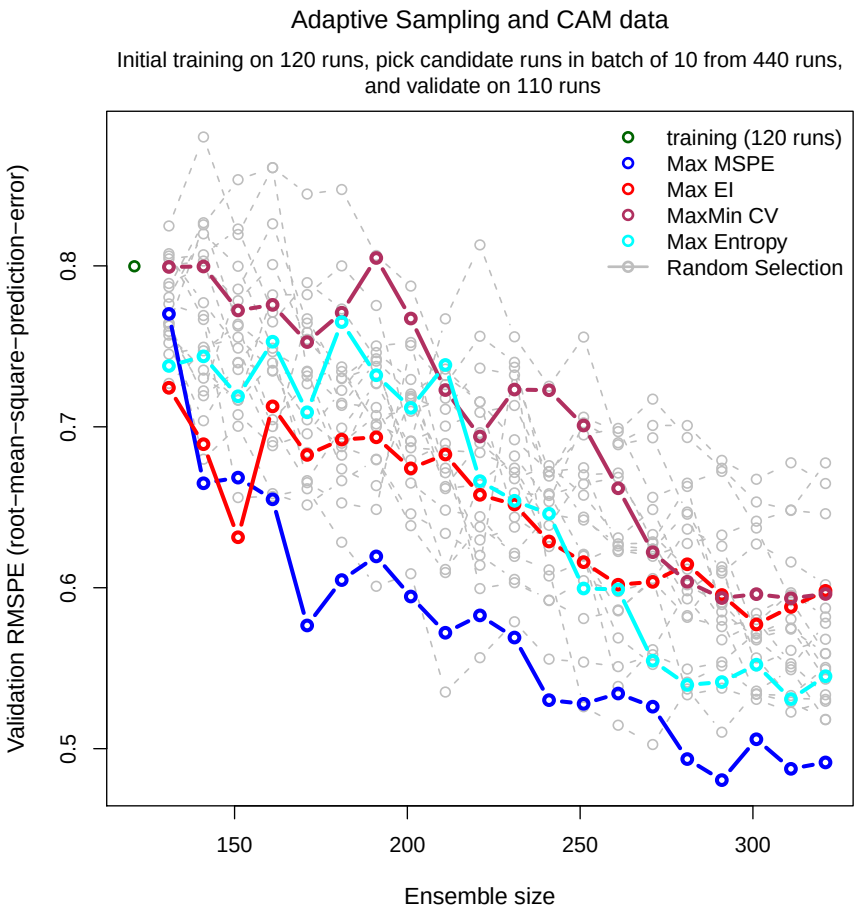

Figure 10: Performance of adaptive sampling on CAM data.

5 Conclusions

Our conclusions may be summarized very briefly: there is no magic bullet among the adaptive sampling criteria we studied. That is, none of them clearly outperforms the others in terms of quickly achieving a global model fit. However, there are several observations we are able to make.

As may be expected, prediction accuracy depends on the dimensionality of the problem. More precisely, this means the accuracy depends on the "actual dimensionality" of the data and not the typically higher dimensionality of the space they are observed in. When the "actual dimensionality" is high, sequential designs tend not to do better than a random selection of runs.

For data from a stationary GP, designs based on MSPE seem to achieve better prediction accuracy than other criteria. This is not too surprising since, in a sense, this criterion is "designed" for Gaussian processes.

For data from a non-stationary process, the expected improvement design typically stands out. One would expect this given that we are approximating data that come from a nonstationary Gaussian process with a stationary GP model and the expected improvement is the only criteria that we study that adapts to the shape of the response model and not only to the distance between design points.

Further work is needed to identify designs that will actively search for "informative" regions in the domain and improve the global fit of the model.

Our findings suggest there is no clear winner in terms of global model fit and our reasoning is that earlier information, from the fitted GP model using fewer observations, for sequential designs, might be misleading in design point selection and hence reduce the effectiveness of the sequential designs.

References

[1] R. A. Bates, R. J. Buck, E. Riccomagno, and H. P. Wynn. Experimental design and observation for large systems. *Journal of the Royal Statistical Society, Series B: Methodological*, 58:77–94, 1996.

[2] W. D. Collins, P. J. Rasch, B. B. Boville, J. J. Hack, J. R. McCaa, D. L. Willaimson, J. T. Kiehl, B. Briegleb, C. Bitz, S. Lin, M. Zhang, and Y. Dai. Description of the ncar community atmosphere model (cam 3.0). Technical report, NCAR, XXXX.

[3] C. Currin, T. Mitchell, M. Morris, and D. Ylvisaker. Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *Journal of the American Statistical Association*, 86:953–963, 1991.

[4] R. Jin, W. Chen, and A. Sudjianto. On sequential sampling for global metamodeling in engineering design., 2002. In Proceedinf of DETC 2002.

[5] D. Jones, M. Schonlau, and W. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 1998.

[6] J. Kleijnen and W. Beers. Application-driven sequential designs for simulation experiments: Kriging metamodeling. *Journal of the Operational Research Society*, 55:876–883, 2004.

[7] J. R. Koehler and A. B. Owen. Computer experiments. *Handbook of Statistics*, 1996. In Ghosh, S. and Rao, C. R., editors, Publisher: Elsevier Science, New York.

[8] C. Q. Lam. Sequential Adaptive Designs In Computer Experiments For Response Surface Model Fit. http://etd.ohiolink.edu/, 2008.

[9] D. V. Lindley. On a measure of information provided by an experiment. *Annuals of Mathematical Statistics*, 27:986–1005, 1956.

[10] J. L. Loeppky, J. Sacks, and W. J. Welch. Choosing the sample size of a computer experiment: A practical guide. *Technometrics*, 51:366–376, 2009.

[11] M. D. McKay, R. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21:239–245, 1979.

[12] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[13] J. Sacks, S. B. Schiller, and W. Welch. Design for computer experiments. *Technometrics*, 31:41–47, 1989.

[14] T. J. Santner, B. Williams, and W. I. Notz. *The Design and Analysis of Computer Experiments*. Springer-Verlag, 2003.

[15] M. Schonlau. Computer Experiments and Global Optimization. PhD thesis, University of Woterloo, 1997.

[16] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 1948.

[17] M. C. Shewry and H. P. Wynn. Maximum entropy sampling. *Journal of Applied Statistics*, 14:165–170, 1987.